UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/066,475	02/01/2002	Edward Colles Nevill	1103179-0009	5943

7470 7590 05/14/2008

WHITE & CASE LLP
PATENT DEPARTMENT
1155 AVENUE OF THE AMERICAS
NEW YORK, NY 10036

EXAMINER	
COULTER, KENNETH R	
ART UNIT	**PAPER NUMBER**
2141	

MAIL DATE	DELIVERY MODE
05/14/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 10/066,475
Filing Date: February 01, 2002
Appellant(s): NEVILL, EDWARD COLLES

Scott T. Weingaertner (Reg. No. 37,756)
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 11/27/06 appealing from the Office action mailed 2/15/06.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The following are the related appeals, interferences, and judicial proceedings known to the examiner which may be related to, directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal:

ARM Ltd. v. picoTurbo, Inc, C.A. No. 4:00-cv-00957-CW (N.D. Cal.)

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

No amendment after final has been filed.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

5,115,500	**LARSEN**	**5-1992**

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 102

1. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

2. Claims 1 – 70 are rejected under 35 U.S.C. 102(b) as being anticipated by Larsen (U.S. Pat. No. 5,115,500) (Plural Incompatible Instruction Format Decode Method and Apparatus).

2.1 Per claim 65, Larsen teaches a program counter register (Fig. 2, item 3 '**IAR**') comprising:

an ordered set of bits (Fig. 2, items 10, 1 (***the top three bits from the IAR combined with the remaining bits from the IAR***); col. 5, line 52 – col. 6, line 2);

wherein a subset of the ordered set of bits (Fig. 2, items 3, 2, 10 (***the bits that are not the top 3 bits from the IAR***)) identifies an address of an instruction (Fig. 2, item 1; col. 5, line 52 – col. 6, line 16); and at least one bit of the ordered set of bits identifies an instruction set (Fig. 2, item 10 "IDSR" (***the 3 bits in the IDSR (top 3 bits from the IAR)***); col. 5, line 52 – col. 6, line 2); and

wherein the at least one bit is not a member of the subset (Fig. 2, items 3, 2, 10 (***the IAR is divided into two sections, the top 3 bits and the remaining bits. The claimed "subset" are the remaining bits of the IAR. As the top 3 bits are not part of the remaining bits, they are not 'a member of the subset'***); col. 5, line 52 – col. 6, line 2).

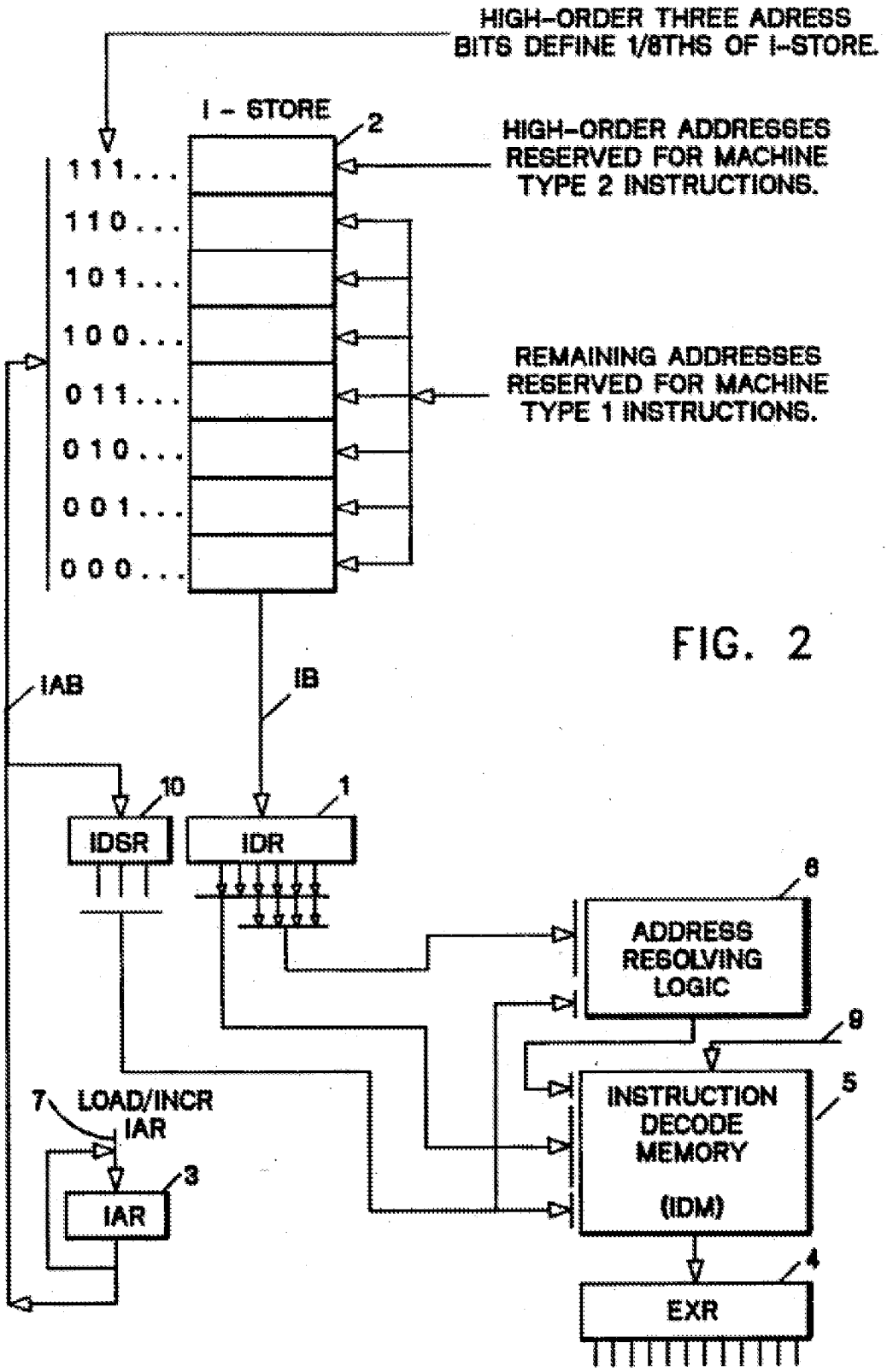

HIGH-ORDER THREE ADRESS BITS DEFINE 1/8THS OF I-STORE.
I - STORE
2
111...
110...
101...
100...
011...
010...
001...
000...
HIGH-ORDER ADDRESSES RESERVED FOR MACHINE TYPE 2 INSTRUCTIONS.
REMAINING ADDRESSES RESERVED FOR MACHINE TYPE 1 INSTRUCTIONS.
IAB
IB
10
IDSR
1
IDR
6
ADDRESS RESOLVING LOGIC
9
INSTRUCTION DECODE MEMORY
(IDM)
5
7
LOAD/INCR IAR
3
IAR
4
EXR

FIG. 2

Col. 5, line 52 – col. 6, line 16 of Larsen

> FIG. 2 illustrates the addition of a new component, the Instruction Decode Selection Register, IDSR, 10. This element is a register that is operated so that it retains the instruction address or some portion thereof of the instruction currently residing in IDR 1. The IDSR 10 is used to trap or retain the address of the last instruction that was fetched from I-store 2. The IDSR 10 retains a portion of the address of the last instruction fetched and will be clocked with the same signal that causes the IDR 1 to be loaded. The retained portion may be all of the address or less if required. Thus, it will always receive **the address of the instruction that is loaded into the IDR 1. In FIG. 2, IDSR 10 is assumed to contain only the high order three bits from the IAR 3** because of the arbitrarily assumed partitioning of I-store 2 in the instant example. **Only the three high order bits necessary to define which of the eight equal arbitrary segments of I-store or partitions are being addressed.**
>
> The contents of IDSR 10 and of IDR 1 are taken together in FIG. 2 to provide a look-up address within the IDM 5. Only specific portions of the undecoded instruction in IDR may be needed according to the format of the instruction being fetched. The other portion of the input address to the IDM 5 is the output from the IDSR 10 which identifies the fetched-from address in I-store 2, or in the instant example, at least the range in which the address lies within I-store 2. Decoding of any specific instruction in the IDR 1 thus depends not only on the contents of the IDR 1 but on the region of the I-store 2 from which the instruction was fetched. This portion of the address is provided by the IDSR 10 output.

2.2 Regarding claim 66, Larsen discloses a method of selecting an instruction set comprising the steps of:

receiving a branching instruction written in a first instruction set of a plurality of instruction sets (Fig. 1A, item 5 "**BRNCH DEC LOGIC**"; Figs. 1A, items 1, 3; Fig. 2; col. 4, line 56 – col. 5, line 12 "a single program which may consist of process instructions or

entire **sub-routines** or code modules from dissimilar machines and in dissimilar or incompatible formats …"; col. 5, line 34 – col. 6, line 2; col. 7, lines 12 – 37);

pursuant to the branching instruction, inserting the address of a next instruction into a register and setting the value of a flag, where the value of the flag is not dependent upon the address of the next instruction (Figs. 1A, 2; col. 5, line 52 – col. 6, line 16; col. 7, lines 12 – 37 "IDR 1 and IDSR 10 will be loaded with the **next instruction to be decoded** and the address in I-store from which it was fetched, respectively.");

selecting an instruction set based upon the value of the flag (Fig. 2; col. 5, line 52 – col. 6, line 16; col. 7, lines 12 – 37); and

acquiring the next instruction at the address inserted into the register (Fig. 2; col. 5, line 52 – col. 6, line 16; col. 7, lines 12 – 37).

2.3 Per claim 67, Larsen teaches a processing apparatus comprising:

a pointer for identifying an address of a next instruction that is written in a first instruction set of a plurality of instruction sets (Fig. 2; col. 5, line 52 – col. 6, line 16; col. 7, lines 12 – 37); and

a flag for identifying the first instruction set (Fig. 2; col. 5, line 52 – col. 6, line 16; col. 7, lines 12 – 37);

wherein:

the pointer and the flag are both written in response to an instruction from a second instruction set of the plurality of instruction sets (Fig. 2; col. 5, line 52 – col. 6, line 16; col. 7, lines 12 – 37), and

the value of the flag is not dependent upon the address of the next instruction (Fig. 2; col. 5, line 52 – col. 6, line 16; col. 7, lines 12 – 37).

2.4 Regarding claim 68, Larsen discloses the apparatus of claim 67 wherein:

the first instruction set is different from the second instruction set (Abstract; Fig. 2).

2.5 Per claim 69, Larsen teaches the apparatus of claim 67 wherein:

the pointer and the flag are located in a single register (Fig. 2; col. 7, lines 12 – 37).

2.6 Regarding claim 70, Larsen discloses the apparatus of claim 67 wherein:

the pointer and the flag are not located in a single register, yet are written to as if portions of a single register (Fig. 2; col. 5, line 52 – col. 6, line 16; col. 7, lines 12 – 37).

2.7 Regarding claim 1, Larsen discloses a data processing apparatus comprising:

(a) a processor core operable to execute successive program instruction words of a predetermined plurality of instruction sets stored in a data memory (Abstract; Figs. 1A, 1B, 2; col. 3, lines 15 – 43);

(b) a program counter register for indicating an address of a next program instruction word in said data memory (Fig. 2, item 3 "**IAR**"; Fig. 1A, item 3 "**IAR**"; col. 3, lines 59 – 62);

(c) logic operable to modify the contents of said program counter register in response to a current program instruction word (Figs. 1A, 2; col. 3, lines 31 – 43; col. 3, line 52 – col. 4, line 6; Additionally, it was well known in the art that a program counter in a microprocessor has a program counter register modifier.);

(d) a processor core controller (Fig. 2, item 5), responsive to one or more predetermined indicator bits of said program counter register (Fig. 2, items 3, 10 (***the top 3 bits from the IAR***)), operable to control said processor core to execute program instruction words of a current instruction set selected from said predetermined plurality of instruction sets and specified by the state of said one or more indicator bits of said program counter register (Fig. 2, items 10, 5, 4; col. 5, line 34 – col. 6, line 2);

(e) a memory access controller operable to access program instruction words stored in said data memory, said access controller not being responsive to said one or more indicator bits of said program counter register (Fig. 2, items 3, 10, 2 (***the bits that are not the top 3 bits from the IAR***); col. 5, line 34 – col. 6, line 2).

Col. 5, line 34 – col. 6, line 2 of Larsen

> FIG. 2 illustrates a block diagram of the major portions of the logic required to accommodate the changes proposed by the preferred embodiment of this invention. The logic shown is similar to that in FIG. 1, but omits the ILR 8 and its loading line 7 for simplicity and includes only the elements necessary for implementing the multiple instruction type decoding of the present invention. In addition, FIG. 2 shows an example of the instruction store 2 with partitioning illustrated for accommodating two types of different, incompatible

format machine language instructions. As illustrated, the high order addresses have been arbitrarily reserved for machine type or format 2 instructions. These are identified as having the high order address beginning with 111 in binary and reserve 1/8 of the instruction store total capacity for machine type 2 instructions. The remaining addresses are arbitrarily reserved for machine type 1 instructions as illustrated.

FIG. 2 illustrates the addition of a new component, the Instruction Decode Selection Register, IDSR, 10. This element is a register that is operated so that it retains the instruction address or some portion thereof of the instruction currently residing in IDR 1. The IDSR 10 is used to trap or retain the address of the last instruction that was fetched from I-store 2. The IDSR 10 retains a portion of the address of the last instruction fetched and will be clocked with the same signal that causes the IDR 1 to be loaded. The retained portion may be all of the address or less if required. Thus, it will always receive the address of the instruction that is loaded into the IDR 1. In FIG. 2, **IDSR 10 is assumed to contain only the high order three bits from the IAR 3** because of the arbitrarily assumed partitioning of I-store 2 in the instant example. **Only the three high order bits necessary to define which of the eight equal arbitrary segments of I-store or partitions are being addressed.**

Col. 3, lines 31 – 43 of Larsen

The instruction address register 3 provides the next instruction address to instruction store 2 over the instruction address bus IAB. Instructions are sequentially fetched and placed in the instruction decode register 1 and portions of the instruction are examined in the address resolving logic 6 which contains logic that defines which field or fields of the instruction are to be used and (perhaps) modified to produce an effective address or direct operand value. In the example shown in FIG. 2, decode tables in the instruction decoding memory or logic 5 may be loaded externally on line 9. This may provide the loading of specific instruction decoding rules as will be later described.

Col. 3, line 52 – col. 4, line 6 of Larsen

In FIG. 1, the instruction decode register 1, abbreviated IDR, is shown. The IDR is the register into which instructions fetched from the instruction store, called I-store 2, are placed to be decoded. Instructions taken from, or "fetched" from, the I-store 2 are placed in the IDR where they are held for decoding. The instructions are stored in the IDR prior to being decoded in the instruction decoding logic or memory 5. The instruction address register 3, abbreviated IAR in FIG. 1, is used to contain the I-store 2 address for the instruction that is to be fetched. The execution register 4, abbreviated EXR, contains decoded instructions which are the execution process codes for the

> specific machine. Instruction decoding logic elements comprising the instruction decode memory or logic as explained earlier, with its contained logic, and the address resolving logic 6 are utilized for decoding the instructions that reside in the IDR 1. The address resolving logic extracts specific bits from the instruction in the IDR, based upon the instruction type, and then causes an effective address or immediate operand value to be calculated by adding (for example) the contents of some index register (R0 or R4 in FIG. 1B).

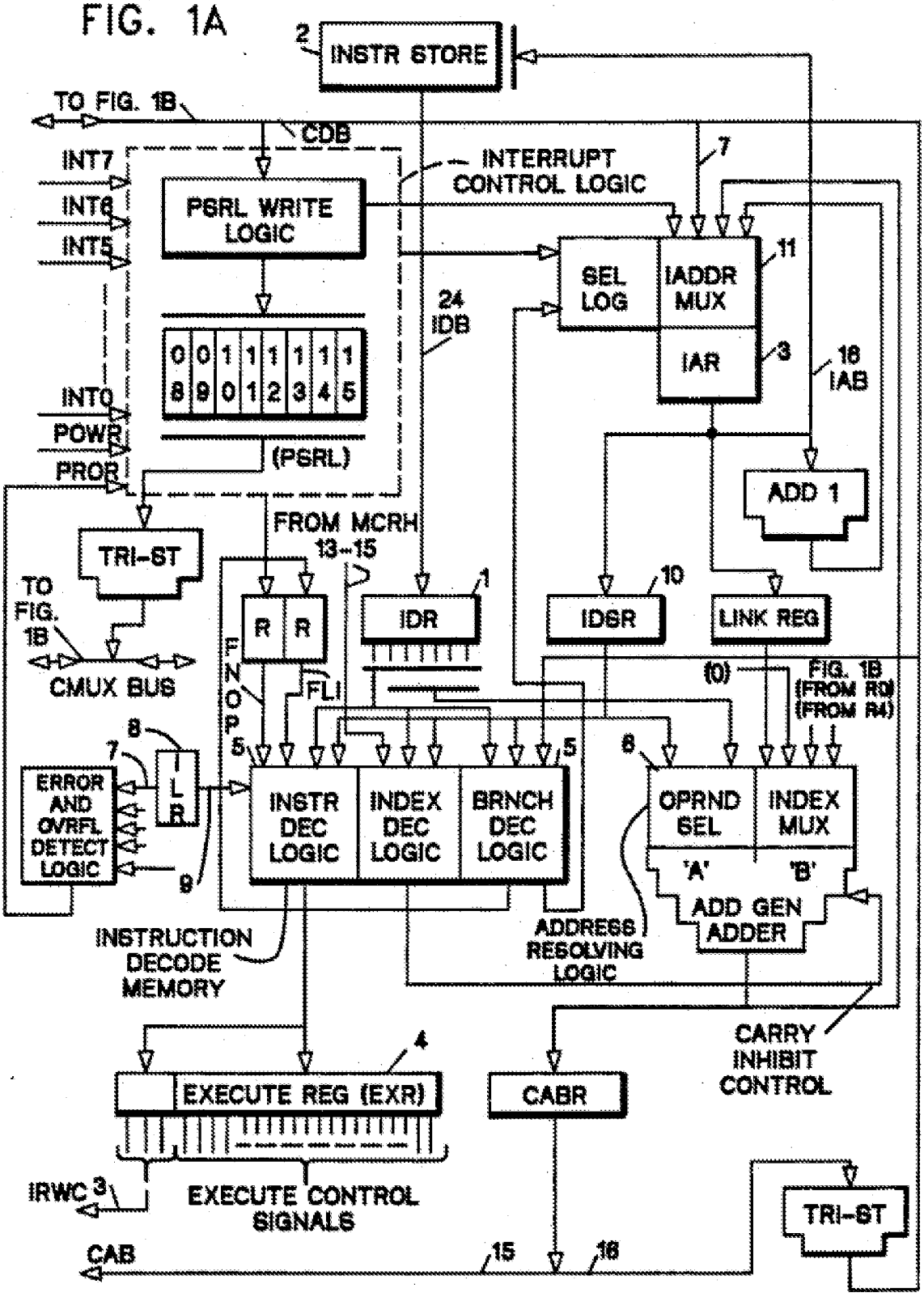
2.8 Per claim 2, Larsen teaches the following additional features:

a first instruction decoder for decoding program instruction words of a first instruction set (Fig. 1A, item 5; Fig. 2, item 5; col. 3, lines 57 – 62; col. 6, lines 3 – 16); and

a second instruction decoder for decoding program instruction words of a second instruction set (Fig. 1A, item 5; Fig. 2, item 5; col. 3, lines 57 – 62; col. 6, lines 3 – 16);

and in which said processor core controller is operable to control either said first instruction decoder or said second instruction decoder to decode a current program instruction word (Fig. 1A, item 5; Fig. 2, item 5; col. 3, lines 57 – 62; col. 6, lines 3 – 16).

FIG. 1A


2
INSTR STORE
TO FIG. 1B
CDB
INT7
INT6
INT5
INT0
POWR
PROR
INTERRUPT CONTROL LOGIC
7
PSRL WRITE LOGIC
24 IDB
0 0 1 1 1 1 1 1
8 9 0 1 2 3 4 5
(PSRL)
SEL LOG
IADDR MUX
IAR
11
3
16 IAB
ADD 1
FROM MCRH 13-15
TRI-ST
TO FIG. 1B
CMUX BUS
R R
IDR
1
IDSR
10
LINK REG
FNOP
FLI
10)
FIG. 1B (FROM RD) (FROM R4)
8
7
5
5
8
ERROR AND OVRFL DETECT LOGIC
L R
9
INSTR DEC LOGIC
INDEX DEC LOGIC
BRNCH DEC LOGIC
OPRND SEL
INDEX MUX
'A'
'B'
ADD GEN ADDER
INSTRUCTION DECODE MEMORY
ADDRESS RESOLVING LOGIC
CARRY INHIBIT CONTROL
4
EXECUTE REG (EXR)
CABR
IRWC 3
EXECUTE CONTROL SIGNALS
TRI-ST
CAB
15
16

2.9 Regarding claim 3, Larsen discloses program instruction words of said first instruction set are X-bit program instruction words (Abstract; Fig. 2; col. 6, lines 41 – 66); and

program instruction words of said second instruction set are Y-bit program instruction words (Abstract; Fig. 2; col. 6, lines 41 - 66);

Y being different to X (Abstract; Fig. 2; col. 6, lines 41 - 66).

2.10 Per claim 4, the rejection of claim 3 under 35 USC 102(b) (paragraph 2.9 above) applies fully.

2.11 Regarding claims 5 and 6, Larsen discloses that Y is 16. (Abstract; Fig. 2; col. 6, lines 41 – 66 "execute also 'type 2' format or language instructions that arbitrarily utilize **16 bit words** …").

However, Larsen does not explicitly disclose that X is a **32 bit** instruction word.

Larsen does teach "**let us suppose** that a 'type 1' format or language instruction set requires **24 bit words** …" (col. 6, lines 41 – 42).

Clearly, it would have been inherent for the 24 bit word example chosen in Larsen to be substituted with a 32 bit word example.

2.12 Per claim 7 , Larsen teaches that said one or more indicator bits of said program counter register are one or more most significant bits of said program counter register (see details of the rejection of claim 1, item (d) above).

2.13 Regarding claim 8, Larsen teaches that said one or more indicator bits of said program counter register are one or more least significant bits of said program counter register (see details of the rejection of claim 1, item (d) above).

2.14 Per claims 9 – 13, Larsen discloses that said one or more indicator bits of said program counter register are one or more least significant bits of said program counter register (see details of the rejection of claim 1, item (d) above).

2.15 Regarding claim 14, Larsen teaches a data memory for storing program instruction words to be executed (Fig. 2, item 4; col. 3, lines 62 – 64 “The execution register 4, abbreviated EXR, contains decoded instructions which are the execution process codes for the specific machine.”; col. 7, lines 15 – 22).

2.16 Regarding claims 15 – 64, the rejection of claims 1 – 14 and 65 – 70 under 35 USC 102(b) (paragraphs 2.1 – 2.15 above) applies.
In addition, Larsen discloses setting one or more control flags to indicate that a current instruction set for the data processing apparatus is the instruction set identified based

on the instruction set indicator portion of the sequence of bits (Fig. 2; col. 5, line 52 – col. 6, line 16; col. 7, lines 12 – 37).

(10) Response to Argument

Arguments regarding claim 65:

Applicant argues that "Larsen cannot have at least one bit in a program counter that identifies an instruction set but is not part of the bits that are used to identify the instruction address because the instruction set in Larsen is directly dependent on the instruction address."

Examiner disagrees.

Larsen meets the limitations of claim 65 as currently written.

Specifically, Larsen discloses a program counter register (Fig. 2, item 3 '**IAR**') comprising:

an ordered set of bits (Fig. 2, items 10, 1 (***the top three bits from the IAR combined with the remaining bits from the IAR***); col. 6, lines 3 – 16);

wherein a subset of the ordered set of bits (Fig. 2, items 3, 2, 10 (***the bits that are not the top 3 bits from the IAR***)) identifies an address of an instruction (Fig. 2, item 1; col. 5, line 52 – col. 6, line 16); and at least one bit of the ordered set of bits identifies an instruction set (Fig. 2, item 10 "IDSR" (***the 3 bits in the IDSR (top 3 bits from the IAR)***); col. 5, line 52 – col. 6, line 2); and

wherein the at least one bit is not a member of the subset (Fig. 2, items 3, 2, 10 (***the IAR is divided into two sections, the top 3 bits and the remaining bits. The claimed "subset" are the remaining bits of the IAR. As the top 3 bits are not part of the remaining bits, they are not 'a member of the subset'***); col. 5, line 52 – col. 6, line 2).

Applicant's actual claim language is broader than what Applicant has argued.

Applicant's claim language does not exclude the top 3 bits from the IAR (the 3 bits in the IDSR) from **both** specifying the address of the instruction and specifying the instruction set in which the instruction belongs.

Arguments regarding claims 66 – 70:

Applicant argues that both independent claims 66 and 67 require "identifying the address of a next instruction and setting the value of a flag, wherein '**the value of the flag is not dependent upon the address of the next instruction**.'"

In addition, Applicant argues that in "Larsen, the bits that designate the instruction set are always part of the address of the instruction; thus, no control flag can be set without regard to the instruction address …".

Examiner disagrees.

Applicant's actual claim language is broader than what Applicant has argued.

Applicant's claim language does not exclude the top 3 bits from the IAR (the 3 bits in the IDSR) from **both** specifying the address of the instruction and specifying the instruction set in which the instruction belongs.

Larsen meets the limitations of independent claims 66 and 67 as currently written. The flag can be considered the top 3 bits from the IAR. The address of the next instruction can be considered the bits that are not the top 3 bits from the IAR. In this scenario, the value of the flag is not dependent upon the address of the next instruction.

Arguments regarding claims 1 – 14:

Applicant argues that "Larsen specifically requires that the instruction set is specified by part of the instruction address (which would be stored in a program counter). Thus the address resolving logic in Larsen must be responsive to the 'indicator' bits of the instruction that specify the instruction set."

In addition, Applicant states that "in Larsen, the bits in the IDSR 10 are always used both to specify the address of the instruction and the instruction set "n which the instruction belongs."

In conclusion, Applicant states in "contrast, independent claim 1 of the present application requires that the memory access controller perform in exactly the opposite manner by 'not being responsive to said one or more indicator bits of the program counter register.'"

Examiner disagrees.

Applicant's actual claim language is broader than what Applicant has argued. Applicant's claim language does not exclude the top 3 bits from the IAR (the 3 bits in the IDSR) from **both** specifying the address of the instruction and specifying the instruction set in which the instruction belongs.

Arguments regarding claims 15 – 20 and 51 – 56:

Equivalent to the arguments with respect to claims 1 – 14.

Arguments regarding claims 21 – 26, 33 – 38, 45 – 50 and 57 – 64:

Equivalent to the arguments with respect to claims 1 – 14.

Arguments regarding claims 27 – 32 and 39 – 44:

Equivalent to the arguments with respect to claims 1 – 14.

(11) Related Proceeding(s) Appendix

Copies of the court or Board decision(s) identified in the Related Appeals and Interferences section of this examiner's answer are provided herein.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

/Kenneth R Coulter/

Primary Examiner, Art Unit 2141

Conferees:

/Richard Ellis/

Primary Examiner, Art Unit 2183

/Nathan J. Flynn/

Supervisory Patent Examiner, Art Unit 2154